REMARKS

Reconsideration of this Application is respectfully requested. A Request for Continued Examination and Petition for Extension of Time are submitted herewith.

Upon entry of the foregoing amendments, claims 1, 3-11, and 13-22 are pending in the application, with claims 1 and 11 being the independent claims. Claims 2, 7-8, 12, 17-18, and 22 have been canceled without prejudice or disclaimer. Support for the subject matter of the amended claims is contained in the application as originally filed.

Applicant submits the foregoing amendments do not raise new issues and present the rejected claims in better form for consideration on appeal.

Based on the above Amendment and the following Remarks, Applicant respectfully requests that the Examiner reconsider all outstanding objections and rejections and that they be withdrawn.

Information Disclosure Statement

An Information Disclosure Statement and the European Search Report from the corresponding European application are filed herewith.

Objections to the Specification

The Examiner made objections to the specification based on grammatical informalities. In response, Applicant submits herewith a substitute specification along with a marked-up copy thereof showing the correction of various grammatical informalities. Applicant respectfully submits that substitute specification includes no new matter.

Rejections under 35 U.S.C. § 112

The Examiner has rejected claims 1, 3-11, and 13-22 under 35 U.S.C. §112, second paragraph as being indefinite. Applicant respectfully submits that the rejection of the claims are overcome or rendered moot by the accompanying amendment thereto.

Rejections under 35 U.S.C. § 103

Claims 1, 8, 11, 18, 21, and 22

The Examiner has rejected claims 1, 8, 11, 18, 21, and 22 under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 5,842,002 to Schnurer et al. ("Schnurer") in view of U.S. Patent Publication 2002/0073055 to Chess ("Chess"). Schnurer and Chess lack the method of the present invention including, *inter alia*, the steps of providing a virtual computer environment, loading objects which may include viruses to the environment, inducing any disruptive behavior of the viruses in the environment, *and* cleaning the viruses from the environment, as is called for by presently amended claims 1 and 11. For example, claim 1 calls for

> A method for detecting and cleaning computer viruses, comprising the steps of:
>
> *simulating in a computer a virtual computer circumstance*, on which virtual computer circumstance the computer viruses will reside;
>
> providing a plurality of objects to be infected by computer viruses that induce virus infection;
>
> *loading a target object to be scanned into said simulated virtual computer circumstance*, said target object being a host possibly attached by a virus;
>
> *activating any virus attached on the target object to be scanned in said simulated virtual computer circumstance to induce virus infection of the target object* and generating standard samples which have been infected, wherein if a virus is attached to the target object and the virus is activated, the target object will include a host body and a virus body;
>
> comparing the plurality of objects after processing in the activating step with the plurality of objects to be infected originally provided, and determining whether there is any change or not, if there is a change, the target object to be scanned contains a virus, otherwise the target object to be scanned is free of viruses;
>
> analyzing the generated standard samples and extracting information on the viruses indicated by changes between the plurality of objects before infection and the standard samples after infection when it is determined that said target object to be scanned contains a virus, said information including at least the size of the virus and key information of the host which has been changed by the virus; and
>
> *cleaning the virus from the infected target object* by locating the host body and the virus body in the target object after the activation step, restoring unchanged key information of the host on the basis of said information, and removing the virus body from the target object after the activation step according to the virus size.

Independent claim 11 calls for a system including means to accomplish substantially the same steps set forth in claim 1, referring to a virtual computer environment instead of a virtual computer circumstance, which for the purposes of the present application a synonymous.

As such, the claimed invention is directed to providing a virtual computer environment, loading a target object into the virtual computer circumstance and activating any virus that may be attached on the target object. When a virus is found, and the activated target object will contain a host body (e.g., the target object less the virus) and a virus body (e.g., the virus). Information about the virus is obtained by analyzing the originally provided target object and standard samples generated after infection. The information includes the sizes of the virus and other key information of the host changed by the virus. Based on such information, the location of the host body and the virus body can be determined and the host body can be extracted from the target object. As such, a cleaned host body can be obtained directly from the original target object that included the virus.

Importantly, the present invention provides for detecting *and cleaning viruses within a virtual computer environment instead of a real computer environment.* As such, there is no risk for the real computer environment to be infected by viruses.

In contrast, Schnurer only discloses a computer virus trap for detecting computer viruses. While Schnurer may provide a virtual computer environment to induce disruptive behavior of a virus, Schnurer fails to teach or suggest how to remove or clean detected viruses from an object within the virtual computer environment. Schnurer discloses a response/alarm means 52 which makes a response or alarm, *see* column 8, lines 27 *et seq.*, however, Schnurer cannot clean the virus and recover the uninfected object.

Chess teaches away from the present invention. Chess is directed to detecting the documents for virus in *a real computer environment* and also removing any virus in *the real computer environment*, which will highly jeopardize the real computer environment. Although Schnurer discloses detecting virus in a virtual computer environment, there is no suggestion or

motivation to combine Schnurer and Chess because the two approaches are contrary to each other.

Chess discloses a method for detecting unknown viruses by recording previous data, such as a document name and macro data in a database, and then checking all the subsequent documents to see if the macro data of the subsequent document are changed. If there is a change, "questionable" changes are identified, and a virus is determined from the "questionable" changes. It appears that , when a virus is found, the repair or removal of the virus is done by user repair or by the method shown in FIG. 3C. In FIG. 3C, the virus is removed by removing all newly emerged macros from the document. In essence, Chess only teaches the method of comparing the document with a previously recorded document, and removing viruses by removing macros which are found for the first time. Such comparison and removal disclosed by Chess is conducted *a real computer environment which includes previous data*, and if there are no previous documents recorded in the database (i.e., step 302) of the real computer environment, viruses cannot be detected and cleaned from subsequent documents.

In particular, in accordance with Chess, there must be a previous document already recorded in the database before any detecting or removing of virus can work. The subsequent document to be detected is a counterpart of the "target object to be scanned" as called for in claims 1 and 11 of the present invention. In the present claimed invention, however, *there is no such need for a previous document or target object* because the system and method claimed does not detect any virus by comparing changes on the target object but instead performs operations on the loaded target object to activate any virus attached on the target object so that the virus will infect the objects provided and cause changes between the objects and the generated standard samples. Therefore, the present invention can work without previously providing a clean target object and is thus superior to the system taught by Chess.

Also, Chess fails to teach or suggest the claimed features of the claimed invention including the extraction of the host body from the target objection by *cleaning the virus from the infected object based on information about the virus including the size of the virus* obtained by comparison of generated standard samples and the original objects. As is known, when a virus

infects an object such as a document (e.g., a so-called host), the virus will attach itself to the host body and may modify some key information on the host body. Although the virus may hide itself in the host, the present invention activates any virus attached to a target object to cause virus infection *in order to identify a host body and a virus body*. *See*, e.g., present application, page 20 lines 1-14. Determining the size of the virus from the comparison of the original target objects and the generated standard samples, the host body can be directly extracted from the target object after activation, and a completely cleaned host body can be obtained. This unique method for obtaining a cleaned host body works together with the above distinctive features in claims 1 and 11 to achieve the advantageous technical effects of detecting and cleaning known and unknown viruses effectively and efficiently *without jeopardizing the real computer environment* and *without the need for a previously-recorded document* as called for by Chess.

Chess fails to teach or suggest the claimed features of the presently claimed invention for extracting the cleaned host body and thus, even in combination with Schnurer, does not teach or suggest the invention set forth in the presently amended independent claims 1 and 11.

For at least these reasons, Applicant respectfully submits that Schnurer and Chess, taken individually or combined, do not render obvious independent claims 1 and 11. Applicant submits that claims 8, 18, 21, and 22, which depend from claim 1 or 11, are allowable over the cited art for at least the same reasons noted above.

Claims 3-6, 9, 10, 13-16, 19 and 20.

The Examiner has rejected claims 3, 4, 13, and 14 under 35 U.S.C. § 103 as being unpatentable over Schnurer and Chess further in view of the publication entitled *Connectix Virtual PC Datasheet and FAQ* ("Connectix"). The Examiner rejects claims 5-7 and 15-17 further in view of U.S. Patent No. 6,067,410 to Nachenberg ("Nachenberg"), and rejects claims 9, 10, 19 and 20 further in view of U.S. Patent Nos. 5,537,636 to Uchida ("Uchida"), 6,356,915 to Chtchetkine et al. ("Chtchetkine") *and* 6,192,456 to Lin et al. ("Lin").

Applicant respectfully submits that claims 3, 4, 13, and 14 , which depend from claim 1 and 11, are allowable over the cited art for at least the same reasons noted above. Connectix

does not teach or suggest, nor does the Examiner rely on Connectix for teachings of providing a virtual computer environment, loading objects which may include viruses to the environment, inducing any disruptive behavior of the viruses in the environment, *and* cleaning the viruses from the environment. As such, Connectix does not account for the deficiencies of Schnurer and Chess.

Similarly, Applicant respectfully submits that claims 5, 6, 9, 10, 15, 16, 19 and 20 are allowable for the same reasons noted above. Nachenberg, Uchida, Chtchetkin and Lin fail to teach or suggest such features and thus do not account for the deficiencies of Schnurer and Chess.

For at least these reasons, Applicant respectfully submits that Schnurer, Chess, and Connectix, taken individually or combined, do not render obvious independent claims 1 and 11. Applicant submits that claims 3-6, 9, 10, 13-16, 19 and 20, which depend from claims 1 or 11, are allowable over the cited art for at least the same reasons. Applicant notes that the rejections of claims 17 and 18 are rendered moot by the cancellation thereof.

CONCLUSION

All of the stated grounds of objection and rejection have been properly traversed, accommodated, or rendered moot. Applicant therefore respectfully requests that the Examiner reconsider all presently outstanding objections and rejections and that they be withdrawn. Applicant believes that a full and complete reply has been made to the outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at the number provided below.

The Commissioner is hereby authorized to charge any underpayment of fees associated with this communication, including any necessary fees for extension of time or additional claims, and/or credit any overpayment to Deposit Account No. 50-2319 (Order No. 469164-00005; Docket No. A-70915/DJB/VEJ).

Prompt and favorable consideration of this Amendment and Response is respectfully requested.

Respectfully submitted,

DORSEY & WHITNEY LLP

Date: April 19, 2006 By: _____
Victor E. Johnson, Reg. No. 41,546

DORSEY & WHITNEY LLP
Suite 1000
555 California Street
San Francisco, California 94104-1513
Telephone: (415) 781-1989 Facsimile: (415) 398-3249

SUBSTITUTE SPECIFICATION
Marked Up Copy

Method and System for Scanning and Cleaning Known and Unknown Computer Viruses, Recording Medium and Transmission Medium therefor

Field of the Invention

This invention relates to software ~~field~~ for scanning and cleaning (that is, detecting and killing) computer viruses, and in particular, relates to a method of scanning and cleaning unknown viruses, and ~~the~~ computer systems implementing the same, and ~~the~~ recording medium and transmission medium for storing and transmitting this anti-virus software.

For a long time, computer viruses have become a big problem harassing people who use computers. Because of the characteristics of infectivity, self-replication and destructibility of the computer viruses, computer viruses have threatened the normal usage of computers, for example, causing loss or modification of data, damaging files, destroying software, and so on. People often use various anti-virus software to scan and clean them.

So far, commonly used anti-virus products can only detect and clean viruses of known types, that is, for various known viruses, the characteristic codes thereof are already known. In this case, the files possibly carrying viruses are scanned and searched for the viruses' characteristic codes. Once a characteristic code is found, it is determined that the file has been infected, thereby goes to clean the viruses. However, this method can not detect viruses of unknown kinds. Only after the new viruses are discovered and analyzed by virus analyzers, can the characteristic codes be acquired. Therefore, the new viruses can't be recognized and detected until the characteristic codes are added into the conventional anti-virus software.

Since the emergence of computer viruses, viruses have been detected by scanning

eigenvalue, that is, when a new virus is discovered, virus analyzers extract one cluster or several clusters of characteristic codes from virus's program body as its eigenvalue, then the virus-scanning software examines whether a file is infected by detecting the presence of the eigenvalue of the viruses in the file. Though anti-virus techniques have made some improvement in the past decade, it is not changed that the method of scanning eigenvalue is the basis of virus-scanning software. The fatal disadvantage of the eigenvalue scanning method (or the virus scanning software) is that only if a virus is discovered and analyzed, and characteristic codes are added into the virus definition libraries by the virus analyzers, can the virus scanning software recognize the virus of the same kind. In other words, the virus-scanning software always falls behind viruses, and the viruses have to be analyzed by the virus analyzers before the software works against this virus.

Conventional anti-virus techniques that can detect some unknown viruses, such as the wide-spectrum scanning method, the heuristic scanning methods, and etc., are based on sufficient classical virus characteristic codes, and some of them run codes of the target object to be scanned for virus on a virtual machine, to empirically judge whether the target is infected and to judge whether the object target has suspicious codes. For example, some domestic or foreign anti-virus companies have developed some methods for scanning unknown viruses, all of which are based on the same idea that they summarize the common ways of viruses' attacking, such as write to disk, and write to files, then scan for these characteristics codes in the target object. These methods actually are of behavior characteristic definitions, and named as inductive virus-scanning methods or heuristic virus-scanning methods. Such methods can detect some unknown viruses and make alarm, but it has poor effect and high rates of misreporting and failing to report. There are two reasons for this. The first one is that the attacking methods of viruses are various and difficult to be enumerated. And the second one lies in that the attacking methods of viruses are legal to the systems in the same way as lots of software tools, thus it is hard to discriminate between them. With

this kind of methods, some of the unknown viruses can be detected and alarmed. However, because of the high rate of misreporting, these methods will bring users unnecessary concerns. And the fatal disadvantage lies in that although it can detect viruses, it is unable to clean the virus. If a target is attacked by a virus, it has to shutdown the computer until the anti-virus software is upgraded. Furthermore, they can not surely determine whether the target objects (files, boot sector, memory and etc.) are infected, but only tells "Possible infected". Up to now, anti-virus products have not been available for cleaning unknown viruses or cleaning known ones without virus characteristic libraries (database or code base).

Summary of the invention

In view of the above problems of conventional anti-virus software, an object of the present invention is to provide a method, system that can effectively ~~detectand~~detect and clean known and unknown viruses and the recording medium or transmission medium therefor. It makes use of the primary characteristic of the viruses, the infectivity, to detect virus' presence so as to solve the problems of detecting unknown viruses effectively. It can detect almost all known and unknown viruses and clean them. It will completely change the situation that viruses could not be cleaned until they are analyzed manually. The present invention can detect and clean unknown viruses in time, so as to greatly reduce the possibility of viruses' damaging to information and data. Skipping manual analysis to most known and unknown viruses, a lot of labor and money will be saved..

The present invention provides a method for scanning and cleaning computer viruses, comprising the steps of: simulating in a computer a virtual computer circumstance that the computer viruses reside; providing a plurality of objects or baits to be infected by computer viruses for inducing virus infection; loading a target object to be scanned into said simulated virtual computer circumstance; activating the target object to be scanned in said simulated virtual computer circumstance to induce the

viruses possibly attached on said target object to infect the plurality of objects to be infected and generating standard samples which have been infected; comparing the plurality of objects after processing in the activating step with the plurality of objects to be infected originally provided, determining whether there is any change or not, if yes, the target object to be scanned contains virus, otherwise the target object to be scanned is free of virus.

The method for scanning and cleaning computer viruses according to the present invention further includes the steps ~~of:analyzing~~of: analyzing and learning from the viruses by analyzing the generated standard samples and extracting information and knowledge on the viruses when it is determined that said target object to be scanned contains a virus; and cleaning viruses from the infected target object by removing the virus's body and modifying key information which has been changed by said virus on the basis of said information and knowledge on the viruses and on the basis of the modification that viruses have made to said infected objects, i.e. the baits.

The present invention further provides a computer system including a general computer for scanning and cleaning computer viruses, comprising: a computer simulation unit for simulating a virtual computer circumstance that the computer viruses resides; a plurality of objects or baits to be infected by computer viruses for inducing virus infection; a control unit for loading a target object to be scanned into said simulated virtual computer circumstance; a virus infection inducing unit for activating the target object to be scanned in said simulated virtual computer circumstance to induce the viruses possibly attached on said target object to infect the plurality of objects to be infected and generating standard samples which have been infected; and a virus decision unit for comparing the plurality of objects after processing in virus infection inducing unit with the plurality of objects to be infected originally provided, determining whether there is any change or not, if yes, the target object to be scanned contains virus, otherwise the target object to be scanned ~~isfree~~is

free of virus.

The system for scanning and cleaning computer viruses according to the present invention further includes: a virus analyzing and learning means for analyzing the generated standard samples and extracting information and knowledge on the viruses when it is judged that there is virus; and a virus cleaning unit for cleaning viruses from the infected target object to be scanned by removing virus's body and modifying key information which has been changed by said virus according to said information and knowledge on the viruses and on the basis of the modification that viruses have done to said infected objects ,i.e. the baits.

The present invention further provides a computer readable recording medium for causing a computer to execute the steps of the method described above for scanning and cleaning computer viruses according to the present invention. Furthermore, the present invention provides a transmission medium for causing a computer to execute the steps of the method described above for scanning and cleaning computer viruses according to the present invention via network transmission.

Brief description to the drawings:

Figure 1 illustrates a block diagram of the framework of the computer system for scanning and cleaning computer viruses according to the present invention.

Figures 2A-2C are the flowcharts of the method for scanning and cleaning computer viruses according to the present invention.

Detailed description to the preferred embodiments:

A computer virus is so named for its infectivity, which is the most essential characteristic of a virus. If a program has infectivity, it is determined to be carrying viruses. Thus it is the most effective method to identify a virus by identifying a

program's infectivity. However, due to the virus' infectivity, to identify the infectivity means to let the virus infect some objects. If this identification is carried out in a real circumstance, it means that the virus is spreading during scanning virus. So it must be carried out in virtual circumstance to verify whether a target object to be scanned (hereinafter "unscanned target object") has infectivity.

The present invention makes use of the infectivity of the viruses, puts the object which has been suspicious to carry virus into a virtual computer circumstance ~~that~~ in which the computer viruses resides and reproduces, activates it and induces it to infect the baits. Further, because various viruses may require certain infection conditions, such as the size and content of the target object, the invention provides all kinds of baits, including bait objects that have different sizes or contents. For example, files like format.com and sort.com are used to induce viruses of DOS COM type. Files like debug.exe and lable.exe are used to induce viruses of DOS EXE type. A floppy disk boot sector, a hard disk boot sector, or a hard disk primary boot sector is simulated to induce viruses of DOS BOOT type. And files like notepad.exe and word.exe are used to induce viruses of WINDOWS PE type, and so on. Different bait objects are used to satisfy viruses' requirements ~~as possible as~~ if they can.

The present invention ~~directs~~ is directed to a new technique for reproducing, detecting or scanning, and cleaning viruses in a virtual computer circumstance, which is a kind of anti-virus method according to behavior- and results. This invention uses a virtual computer circumstance to simulate a real computer circumstance, in which all the processes of viruses' replication and spreading are realized. At the same time the reproducing and spreading procedure of the virus is monitored, the virus's methods of infection are learned, and the converse process of the infection can be deduced, which forms the method of cleaning such viruses. Below are the detailed steps. First, a virtual circumstance is established in which the virus resides and reproduces, and the suspicious ~~target object to be scanned~~unscanned target object is put into the virtual

circumstance. Second, the suspicious object is activated. If it really carries a virus, the virtual circumstance will be an infected one. Then various operations are performed to the baits in the virtual circumstance to induce the viruses to infect them ~~as possible as it can~~ if they can. In other words, an experiment of the virus's replication and infection is done in the virtual circumstance. If the baits are infected by the virus, the ~~target object to be scanned~~unscanned target object does carry a virus and the baits infected by the virus become the standard samples. Third, if the previous step of replication and infection experiment succeeds, then the standard samples will be analyzed by a program instead of a virus analyzer, and information required for scanning and cleaning the virus is extracted from the standard samples. Fourth, the information obtained from analyzing the standard samples with said program is applied to the infected target object which carries the virus, to clean the virus.

Figure 1 illustrates the block diagram of the computer system for scanning and cleaning computer viruses according to one preferred embodiment of the present invention. As shown in figure 1, a general computer system 1 contains a virus scanning-cleaning unit 2 according to the present invention that can be executed by the computer. The computer system 1 include a general CPU, a memory, an Operation System (OS), a peripheral storage devices (hard disk, floppy disk, and so on)(not shown in figure 1). The whole program of the virus scanning-cleaning unit 2 is executed by the CPU in computer system 1. The computer system further includes a ~~target object to be scanned~~unscanned target object 19, which may be a file in the hard disk or floppy disk or a boot sector of the hard disk or floppy disk in the computer system 2, and even files and data downloaded and transmitted through the Internet that possibly carry a virus.

As shown in Figure 1, the virus scanning-cleaning unit 2 includes a virus scanning control unit 3 for inputting the ~~target object to be scanned~~unscanned target object 19 to the simulated computer circumstance and for controlling the processes in all the

virus-scanning components; a computer simulation unit 4, i.e. a virtual computer, for creating a whole simulated computer system as a virtual circumstance or environment in which the viruses replicates and spreads, the created computer system may include a virtual CPU 5, a virtual memory 6, a virtual Operation System (OS) 7, virtual peripheral storage devices (hard disk, floppy disk, and so on) 8, and other portions of system resources 9 required for the virus's living, replication and spreading, such as system time; one or more standard baits (i.e. the target object possibly to be infected by the ~~computer~~computer viruses) for inducing virus infection; an virus infection inducing unit 10 for loading the ~~target object to be scanned~~unscanned target object 19 into said virtual computer 4 and performing operation, and using the standard baits 11 to cause the viruses possibly carried by the ~~target object to be scanned~~unscanned target object 19 to infect the standard baits 11 and the boot sectors of the virtual hard disk, floppy disk and the like of the simulated computer circumstance, thus generating infected standard samples 13; a virus comparison-decision unit 12 for checking whether the boot sectors of the virtual hard disk, floppy disk or the like of the simulated computer circumstance is changed before and after the step of virus inducing, and comparing the standard samples after infection 13 with the standard baits 11 before infection to determine whether there is any change or not, if yes, it is determined that the ~~target object to be scanned~~unscanned target object 19 contains virus, otherwise there is no virus 18.

The virus-cleaning part of the virus scanning-cleaning unit 2 includes a virus-cleaning control unit 17 which is used to control the processes in all the virus-cleaning components; a virus analyzing and learning unit 14 for analyzing the modification by the virus's infection according to the standard baits 11 and the infected standard samples 13, and to learn knowledge about the virus; and a virus-cleaning unit 15 for killing or cleaning the virus adaptively on the basis of the knowledge obtained from the virus analyzing and learning unit 14 and generate the object 16 from which the virus has been cleaned. The cleaned object 16 can be used to overwrite the input

~~target object to be scanned~~unscanned target object 19 by the virus-cleaning control unit 17 to eliminate the virus.

According to one embodiment of the present invention, the virus-scanning control unit 2 and the virus-cleaning control unit 17 may be integrated as one single control unit to monitor all the above mentioned virus scanning and cleaning processes.

The virtual computer circumstance created by the computer simulation unit 4 includes the virtual machine 5_(virtual CPU), the virtual operating system 7, the virtual peripheral computer storage devices 8, the virtual physical memory 6, and so on. In a word, all the computer resources required for the virus's residence will be simulated; objects possibly carrying viruses are those that may have been infected by viruses theoretically. Objects possibly carrying viruses are put into the virtual circumstance and activated under appropriate conditions.

The virtual CPU 5 may also be called as softcpu() (a CPU implemented or simulated by software). A softcpu() is an interpreter for real CPU instructions. It interprets and executes the program like a real CPU, and it can understand each line of codes, interpret and execute them correctly. Theoretically, the softcpu() can execute all the codes that a real CPU can ~~do~~execute, or interpret and execute all the programs that a real CPU can ~~do~~; it can recognize all the instructions that a real CPU can ~~do~~, or act the same as a real CPU under any statuses. All objects the real CPU operates on (such as BIOS chips and disks) are real ones, while those the virtual CPU operates on (such as BIOS chips and disks) are virtual ones.

In addition, the softcpu() is just a function interpreting the instructions of a real CPU, which may be written in the assemble language, the C language or other languages. In one embodiment of the present invention, it is written in the C language for consideration of portability and maintainability.

The softcpu() would simulate an Intel's CPU if a virus in an Intel computer is to be scanned; it would simulate a Macintosh CPU if a virus in a Macintosh computer is to be scanned.

All programs run in a specific operating system, so do viruses. Virtual operating system 7 ~~is to~~ simulates the operating system on which the virus typically runs. The virtual operating system 7 may include multiple operating system required for viruses' running, such as a virtual operating system for DOS, a virtual operating system for WINDOWS 95, or a virtual operating system for UNIX, and so on. To enhance the efficiency, in one embodiment of the present invention, the virtual operating system 7 only simulates the necessary kernel of the operating system for running the viruses. For DOS viruses, the virtual operating system for DOS would be selected; for Windows 95 viruses, the virtual operating system for WINDOWS 95 would be selected.

The computer simulation unit 4 according to this invention creates virtual peripheral computer storage devices 8, such as hard disks or floppy disks. In the virtual computer circumstance, all writing or reading to the peripheral storage device in the program of the object to be scanned are to the virtual ones, which means, the infection and damage to files and data in the disks caused during the virtual program running are limited to infection and damage to files and data in the virtual disks.

In one embodiment of the present invention, the ~~said~~ virtual computer peripheral storage device 8 includes a function or program unit 8 called by the computer simulation unit 4, to create a virtual hard disk. The primary function of the virtual computer peripheral storage device 8 is to assign an area of required size in the memory, and simulate, according to specific requirements, a virtual hard disk in the memory area, which has the same structure as a normal one, such as having

three-dimension space by sector number, track number and cylinder number, the primary boot sector and its corresponding blank sector of the 0 track, and next the boot sector, the File Allocation Table (FAT), the root directory area, and the necessary system files (i.e., IO.SYS, MSDOS.SYS, COMMAND.COM are required for the DOS system), as well as bait files for testing (i.e., files like DOSEXE.EXE, DOSCOM.COM are required for viruses of DOS files type). The data in the virtual hard disk, which are useful for the scanning-cleaning system of the invention, only occupy an memory area of a size from tens of kilobytes to several hundred kilobytes, while a normal hard disk has a capacity from several megabytes to several gigabytes, most of which are not used in the system according to the present invention. So in one embodiment of the present invention, to simulate a hard disk with the size from several megabytes to several gigabytes, only a block of memory of size from tens of kilobytes to several hundred kilobytes is required. Since only a little memory is required to simulate a hard disk of large capacity, the hard disk required for this system may be realized on a general purpose computer. Furthermore, because the real hard disk will not be accessed during the period of scanning and cleaning~~,~~ and the virtual hard disk is actually in a small area of the memory, the processing speed will be high and time will be saved. In addition, the virtual hard disk is just a part of memory, the real disk will not be infected nor damaged, the physical characteristics of the memory will not be destroyed, so it is harmless to the user's system.

According to one further preferred embodiment of the present invention, when unit 8 is used to simulate a hard disk, a global structure variable Hard_Disk_Struct may be predefined to control the specification of the simulated hard disk, such as a blank disk, a boot disk, a disk containing system files and bait files.

The ~~said~~ virtual device 8 can also simulate a floppy disk primarily by assigning an area of memory of required size and configure the virtual floppy disk in the memory area to have the same structure as a normal one, such as having the boot sector, the

File Allocation Table (FAT), the root directory area, the necessary system files (i.e., IO.SYS, MSDOS.SYS, COMMAND.COM for DOS system), and bait files for testing (i.e., files like DOSEXE.EXE, DOSCOM.COM)–, for which all data required just occupy a size of tens of kilobytes. In one embodiment of the present invention, an global structure variable, floppy_disk_struct, may be predefined to control the specification of the virtual floppy disk, such as a blank disk, a boot disk, a floppy disk containing system files and bait files, for example, floppy disks of sizes of 360 kilobytes, 720kilobytes, 1.2 megabytes, 1.44 megabytes may be created according to the global variable.

In the same way, hard disks or floppy disks of any other operating systems may be simulated. The above flexible implementation will decrease the consumption of the system time, and in case of being called, the virtual peripheral device 8 uploads required data to the designated memory area.

All the above program units-, including the virtual CPU 5, the virtual memory 6 and the virtual OS 7, can be realized by a person skilled in the art with known programming languages. They includes all kinds of instructions to simulate a CPU, all kinds of management and accessing operations to the memory, and all kinds of data structures and implementing codes of function services of the operating system, all of which may be implemented by available programming techniques, so the details thereof are omitted.

To activate the ~~target object to be scanned~~unscanned target object is to activate the virus contained in the target object to behave as a virus. For example, if the target object is an executable binary file (a DOS EXE file, a DOS COM file, a DOS BAT file, or a WINDOWS NE or PE file), to activate means to execute; if the target object is a document file such as a WORD file with executable macros, then to activate it means to open it in the way that the macros can be executed.

The above standard bait may also include setting a virtual system time, including various date and time, to induce viruses which are sensitive to date and time, such as the CIH virus (attacking on April 26), the Friday 13th, and so on. As illustrated in Figure 1, the scanning part of the virus scanning-cleaning program 2 provides a set of standard baits, which includes several standard baits 11 or bait sets. The baits refer to the known objects that are possible to be infected by viruses. In one embodiment of the present invention, the baits are DOS programs for DOS viruses-; WINDOWS 95 programs for WINDOWS 95 viruses; WORD documents for WORD viruses; and so on. The baits are executable entities of the same type as the target objects whatever the target objects are. The baits are clean and all sizes, content, structures and behavioral functions thereof are known, while whether the target object carries a virus is unknown before it is scanned. So their sizes, content, structures and behavioral functions are unknown if they really carry viruses.

In addition, the above baits 11 can not be selected freely, but must be executable entities that may be infected by many known viruses tested by lots of experiments for known viruses. Their sizes, content are "delicious" for viruses, that is, they are apt to be infected. If the baits are infected by the virus, information can be extracted therefrom. In a word, baits are known executable entities apt to be infected, and the bait set is a set of known executable entities of all kinds which are apt to be infected.

Specifically, according to one embodiment of the present invention, the standard baits 11 are configured to include, for example, a bait set of DOS COM type which includes a plurality of bait files which should have different sizes from 1 kilobytes to 60 kilobytes (1 kilobytes, 2.5 kilobytes, 12 kilobytes, 20 kilobytes, 30 kilobytes, 40 kilobytes, and so on); the first instruction of the files in the bait set should be JMP, CALL, MOV, and XOR respectively; the files in the bait set also should have different time, date and attributes to induce viruses of different types, which are

sensitive to them.

The above standard baits can be configured to include a bait set of DOS EXE type, which include some bait files, of which the file headers have sizes of 0x20, 0x200, 0x400, 0x600, or 0x800; and which have sizes of 4 KB, 10 KB, 20 KB, 40 KB, or 80 KB; and of which the last pages have sizes of 0x00, 0x03, 0x80, 0x87, 0x100, or 0x198; and of which the numbers of relocation items are 0x00, 0x01, 0x02, 0x04, or 0x10, respectively, but do not completely occupy the relocation item table; and of which the CS and IP registers should have various values; and the stack's location of the program body can be, for example, at the head, in the middle, at the tail of the program body, or next to the program body (out of the program body).

The above standard baits can be configured to include a bait set of boot type that include sets of boot sectors or primary boot sectors of different versions for MSDOS, PCDOS, DRDOS, WIN9X systems. Actually, they are the virtual hard disks or floppy disks containing boot sectors or primary boot sectors of different versions for MSDOS, PCDOS, DRDOS, or WIN9X to induce viruses of BOOT type created by the computer simulation unit 4.

Similarly, the above standard baits can be configured to include bait sets for MACRO viruses that include WORD documents of various sizes and types to induce MACRO viruses to infect.

As illustrated in Figure 1, the virus infection-inducement unit 10 (also called as virus sample-creating machine) is a function unit that uses all kinds of above bait sets to perform the process for inducing infection of viruses, that is, to run the files to be scanned and the possible viruses attached therein so as to let the standard host files (i.e. all the above baits) be infected by the viruses ~~as possible as~~if it can. Then the virus-identifying unit 12 determines that if there are any baits infected in virus

sample-creating machine 10. Specifically, the virus-identifying unit 12 compares the baits after running the ~~target object to be scanned~~unscanned target object in virus infection-inducing unit 10 with the respective baits before the running, to examine for any change. If there is any bait change before and after the running, the target object is determined to carry a virus, and the changed bait becomes a virus sample. In other words, if the virus sample-creating machine 10 has not created the samples 13, the target objects is clean; otherwise, the target object carries a virus and the standard host file (the bait) becomes a standard sample, which contains all information for cleaning the virus. According to one embodiment of the present invention, if a DOS virus resides in the above virtual memory of the above virtual DOS system, the virus sample-creating machine 10 operates on the baits of the DOS EXE, DOS COM types by executing, opening, reading, closing, or searching, etc., to induce the virus in the memory to infect the baits as possible as it can. In case of being modified or infected, the target object becomes a standard sample 13.

For viruses of the document type, the bait files having been infected are standard samples themselves. But for viruses of the boot sector type, the virus sample-creating machine 10 creates the standard samples according to the boot sector information of the virtual hard or floppy disks, which have been changed by the virus.

The ~~said~~ standard sample 13 refers to the standard bait or host that has been infected by the virus. A standard host is an executable body with known size, content and structure known by virus analyzers, which is suitable for carrying a virus under appropriate infection conditions.

As illustrated in Figure 1, in one embodiment of the present invention, the virus-learning machine 14 in the virus cleaning part according to the invention (also called as standard sample analyzing machine) compares the above standard baits 11 with the created standard samples 13, analyzes the samples, and extracts all

information for the virus or information required to clean the virus. This process is called as the learning process of the virus-learning machine. The learning process of the virus-learning machine is a virus-cleaning process by simulating manual work, and does not use characteristic codes, which is completely different from those that clean viruses using characteristic codes. The information or knowledge picked up by the virus-learning machine from the standard samples includes: the virus's size; the virus's location in the file host; whether the virus is encrypted and transmuted; whether the virus has encrypted the host; whether the virus has damaged the host program too greatly to be cleaned (can only be deleted); whether the virus has relocated the host; whether the virus has aligned the segments of the host; and whether the value or location of key information (such as the entrance of the host program) of the host object have been modified.

For example, for viruses of normal DOS COM types, the virus-learning machine 14 extracts two pieces of knowledge, the first is the virus's size and the second is whether the original functions of the host object have been kept integral or damaged by the virus. One of the algorithms used to calculate the virus's size is to subtract the size of the standard bait (or host) from that of the standard sample; the algorithm for determining whether original functions of the host object ~~is to run~~ runs the standard sample in the virtual computer environment created by the computer simulation unit 4 until it is over or until the virtual computer is down. If the original functions of the standard bait appear during the process, the original functions of the host object are integral, otherwise they have been destroyed..

Traditional virus-cleaning methods using the characteristic codes cleans the virus according to the information (data or codes) in the characteristic libraries of known viruses, which have been filled in by virus analyzers. But the virus-cleaning unit 15 ~~according to this~~of the invention is a virus-cleaning unit which simulates manual work and cleans viruses according to the knowledge ~~real time~~ learned by the virus-learning

machine 14 in real-time without characteristic code libraries of known viruses. The principle ~~to~~ for cleaning virus is "Who ties, who unties", that is, the virus sample-creating unit 10 and the virus-learning unit 14 learn the virus's infection process and analyze the infection results (the standard samples) to acquire the virus's data and attributes~~; and a virus's nature lies in~~. As a characteristic of a viruses is the ability of camouflaging themselves while infecting and promulgating, ~~that is to say,~~ most viruses will not damage the original functions of the host, and if the virus-cleaning unit 15 executes the virus's virtually, the virus will restores the host object, and the virus-cleaning unit can save the host object restored by the virus to disks as the cleaned object (Corresponding conversions should be made if the object exists in different manners in the memory from that in the disks). ~~When~~ Whether the virus restores the host object must be judged according to the virus's attributes learned by the virus-learning machine. For example, in one embodiment of the present invention, the method of self-restoring by the virus is one of the methods by which the virus-cleaning unit 15 deduces the converse process of the infecting, i.e., the process of cleaning virus. If the virus-learning machine has learned enough data or attributes of the virus, it calculates the key information (information modified by the virus) of the original host with all the attributes or data so as to clean virus. Real-time learning of the virus and real-time cleaning ~~it~~ the virus with previously learned knowledge ~~is realized~~ by the virus-cleaning unit 15 is not advantageous over all known anti-virus software products.

In one embodiment of the present invention, the processes ~~that~~ in which the virus-cleaning unit 15 cleans a normal virus of DOS COM types are as follows: firstly, if the original functions of the standard sample are not integral, the files carrying a virus will be deleted, otherwise the process goes to the next step; secondly, the target file of DOS COM type to be scanned is loaded into the virtual computer circumstance and executed until the value of the program segment register CS in the virtual CPU equals the address of the program segment prefix register and the value of the IP

register is 0x0100; thirdly, calculating the size of the cleaned target file-, the size of the cleaned DOS COM file is calculated by subtracting size of the virus from that of the infected DOS COM file; fourthly, generating the cleaned target DOS COM file by saving the content of the virtual memory from CS:IP to CS:IP + size of the cleaned DOS COM file as a file.

If the above virus-learning machine 14 fails to obtain knowledge about the virus or if the virus-cleaning unit 16 determines that the original functions of the host have been damaged, the target object would be deleted.

Figure 2A, 2B, and 2C illustrate the processing flow chart of the virus-cleaning method according to one embodiment of the present invention. All steps of this flow chart are executed in the respective processing units of Figure 1 to form a whole virus scanning and cleaning process. As illustrated in Figure 2A, at first, the ~~target object to be scanned~~unscanned target object 19 is read from the data inputted from the hard disk, the floppy disk or the Internet(step S101), then judgment is made about whether the target object is an object possibly carrying virus (step S102). An object possibly carrying virus is available to be infected by a virus theoretically, but not necessarily carrying a virus. The object possibly carrying virus must be an executable entity, such as *.exe, *.com, *.bat, *.doc, files of NE or PE types, and boot sector or primary boot sector of a disk, and so on. An entity which can not be executed is impossible to carry a virus, such as *.txt.

If in the step S102 the object is determined to be an object possibly carrying virus, the process goes to the step S103 to scan and clean the virus; if the target object is impossible to carry virus, for example, an object that can't be executed like *.txt, the target object is determined to be clean; if the target object is unknown, the object is reported to be an unknown one.

In ~~the~~ step S103, the computer simulation unit 4 creates a virtual computer circumstance, which includes the virtual CPU, the virtual OS, the virtual peripheral storage devices (a hard disk or floppy disk), the virtual memory, and virtual system time, so as to virtually execute the object possibly carrying virus therein. And in ~~the~~ step S104, a plurality of baits are provided which may be infected by the virus (the standard baits 11 in Figure 1), including the above bait set of file types and the bait set of boot sector type for the virtual hard disk or floppy disk. In ~~the~~ step S105, the target object 19 are loaded into the virtual computer circumstance. In ~~the~~ step S106, the virus possibly attached on the target object is activated, that is, induced to infect the virtual computer circumstance and the bait files. On the one hand, in ~~the~~ step S107, judgment is made whether there is any bait having been infected. On the other hand, in ~~the~~ step S108, judgment is made whether the virtual computer circumstance has been infected, that is, if the virtual memory, the boot sector of the virtual hard disk or the virtual floppy disk has been infected. If in step S107, a bait are judged to have been infected, the process goes to ~~the~~ step S111 in Fig. 2B; otherwise, the target object is reported to be clean. If in ~~the~~ step S108 the virtual circumstance is determined to carry virus, ~~on~~ the process goes on to ~~the~~ step S110 in Figure 2B, and as many as possible operations are performed on the baits in the virtual computer circumstance to induce the virus to infect them ~~as~~ if possible ~~as it can~~. After that, the process goes back to S107 to judge again whether there is any bait having been infected.

As shown in Figure 2B, in ~~the~~ step S111 it is reported that the ~~target object to be scanned~~unscanned target object ~~does carry~~ carries a virus, the standard samples are created, and the virus's type are analyzed, such as a DOS virus, a MACRO virus, or a boot sector virus. Then the process goes to ~~the~~ step S112 to prompt the user to decide whether the virus should be cleaned. If the user does not need to clean the virus, a report that the target object is infected, and in ~~the~~ step S109, the scanning process is over. Otherwise, if the user needs to clean the virus, the process goes to step S113.

In ~~the~~ step S113, all the standard samples created in the virtual computer circumstance are extracted. And in ~~the~~ step S114, these extracted standard samples are analyzed by the virus-learning machine 14, in which it is determined ~~the main part is to judge~~ whether the original functions of the standard samples (the functions of the standard bait) have been changed. In ~~the~~ step S115, the integrality of the standard host's original functions (functions before the host is infected) is examined. If it is not integrated, the process goes to step S116. Otherwise, the process goes to step S120.

In ~~the~~ step S116, if the original functions of the host have been damaged by the virus are too great to be restored, the host will have to be deleted. In ~~the~~ step S117, the user is inquired if he wants to delete the infected file. If YES, the file is deleted (step S117); otherwise, the virus-cleaning process is ended (step S119).

As illustrated in Figure 2C, in ~~the~~ step S120, the virus-learning machine 14 learns all knowledge concerning the virus and obtain the key data or attributes required for cleaning the virus as possible as it can, until it learns enough. For example, for a DOS COM virus, knowledge of the following sequence is enough: first, the virus is not encrypted, nor transmuted, and its size is not variable; second, the virus's size virus-size; third, the virus has only changed the first three bytes of the host; fourth, the location where the virus places the first three bytes of the host.

Then in ~~the~~ step S121, the virus-cleaning unit 15 searches or calculates the key data or attributes modified by the virus in the host to be scanned (the host object), based on the knowledge learned by the virus-learning machine 14. For example, for a DOS COM virus, the following information is knew: first, the virus is not encrypted nor transmuted, and its size is not variable; second, the virus's size; third, the virus has only changed the first three bytes of the host; fourth, the location where the virus places the first three bytes of the host has been known(data_offset_in_virus)(relative

to the virus body). Then the steps for cleaning the virus are as follows: first, the location of the virus's body in the file(virus_offset_in_file) is calculated by subtracting the virus's size(virus-size) from that of the target file(file_size); second, location of the first three bytes of the host(data_offset_in_file)(relative to the host file) is calculated, which equals the sum of virus_offset_in_file + data_offset_in_virus.; third, the first three bytes of the host file is replaced by the three bytes data at the location of data_offset_in_file; fourth, the last portion of the infected file is cutoff from the tail by virus_size bytes..

In ~~the~~ step S122, a judgment is made whether the calculation of the original value of the host's information modified by the virus is successful. If not, the virus-cleaning process fails (step S125); otherwise, the process goes to step S123.

In ~~the~~ step S123, the data or attributes of the target file (the host object) which were modified by the virus are restored, such as the size of the file, and the file header data. Thus, the virus is cleaned.

In ~~the~~ step S124, a report is given that the virus has been cleaned successful. Then the process goes to ~~the~~ step S119 and the virus-cleaning process is ended.

The above virus scanning-cleaning method according to this invention and all the respective units can be realized using normal computer programming languages (such as the C language) to program corresponding software, and the software may be executed in a local computer; or may be stored in the floppy disks so as to be sold or used; or may be transmitted or downloaded through networks or the Internet, and then be executed.

The computer system and the method for scanning-cleaning computer viruses realized by software according to this invention can make use of the basic characteristic of

computer viruses, the ability of infection, to detect the virus and real-time learn and use the knowledge about the viruses, which is advantageous over all conventional anti-virus software products. This invention identifies a virus according to its "result" instead of its specific behaviors, so it can be named as the technique according to behavior-result. Of course, the method of this invention also knows both the virus's behaviors and its results of these behaviors, according to which it can safely clean the virus. But it does not examiner for specific individual behaviors (such as writing to the disk), so it can save much system time, and has a higher speed. Further, this invention only uses a small area of the real memory to provide the virtual circumstance for the virus's residence and replication, so it has rapid enough processing speed to realize inducement of the virus to infect as possible as it can.

With the computer system and the method according to this invention, most known and unknown viruses will no longer require manual analysis and can be cleaned without the virus characteristic libraries; it can find new emerging viruses in time; the numbers of the viruses it can clean is limitless; and the anti-virus software using this invention will no longer fall behind the viruses, and it can reliably detect and clean unknown viruses.

While the invention has been particularly described with respect to preferred embodiments thereof, it is no meant to limit the scope of the invention. It will be understood by those skilled in the art that various variations or modifications in details may be made without departing from the scope and spirit of the invention. Thus the scope of the invention is to be defined by the attached claims.